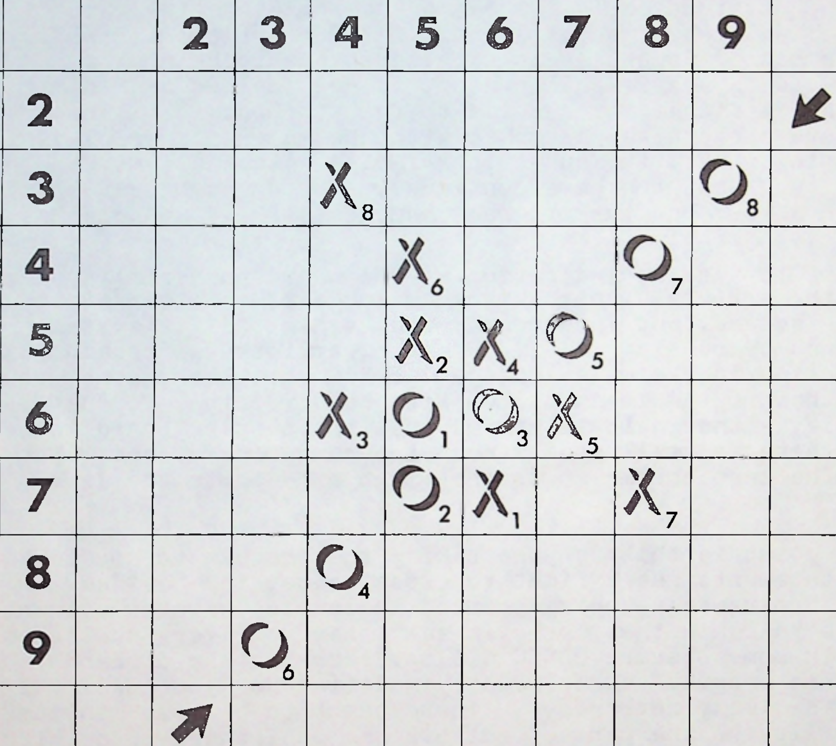
83



Popular Computing

The only magazine for people who enjoy computing.

Volume 8 Number 2

February

1980

In the October 1979 issue of Personal Computing, there appeared a program (in IMSAI BASIC) to play Gomoku. The program was apparently written by Jerry Crouch.

The game of Gomoku (sometimes called Go-Bang, or Fives) itself was described in our issue 49. It is essentially Tic-Tac-Toe on a large grid, with the objective of getting 5 in a row in any direction. Some definitions of the game call for a 19 x 19 board (as in the game of Go); some call for an infinite grid. The distinction is largely academic, since few games ever exceed an area of about 12 x 12. Our article (in April, 1977) recommended Gomoku as an excellent game to be programmed. The program being reviewed, however, is not what we had in mind; it would be difficult for a child to fail to beat it. The cover Figure shows a sample game, with the human player playing O and playing first; the subscripts indicate the half moves. O wins at (9,3) but the game continues. By the 8th half move, O has 7 in a row, but the program then moves (3,4) and claims a win.

There is also no indication of the board numbering system (indeed, the article with the program suggests a letter/digit coordinate scheme, but the program uses a digit/digit system, as one finds by running it). The program looks as though it would function in a simple (integer) BASIC, but this is not true. Further, the program is furnished with no directions for use. For example, although it is stated that the playing board is 10 x 10, there is still doubt about the numbering system; that is, does the zero column go to the right of 9 or to the left of one?

The point is this: anyone taking the trouble to key in a hundred statements has a right to expect something besides trivia. Programs are being printed wholesale, with no evident attempt to validate them, or even edit them. Every published program (in some obscure BASIC dialect) carries the offhand remark "This program can be readily modified for your BASIC," and this is simply not true. There ought to be some minimum responsibility on the part of editors and publishers to certify, validate, or at least run the programs being published.

Publisher: **Audrey Gruenberger**
Editor: **Fred Gruenberger**
Associate Editors: **David Babcock**
Irwin Greenwald
Patrick Hall
Contributing Editors: **Richard Andree**
William C. McGee
Thomas R. Parkin
Edward Ryan
Art Director: **John G. Scott**
Business Manager: **Ben Moore**

POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91302. Subscription rate in the United States is $20.50 per year, or $17.50 if remittance accompanies the order. For Canada and Mexico, add $1.50 per year. For all other countries, add $3.50 per year. Back issues $2.50 each.

More on Powers of 2

The function $F = 2^X$ seems to fascinate mathematicians. We devoted an entire issue (number 69) to these two problems:

1. Given a positive integer, R, find the smallest X such that 2^X has its R low-order digits all 1's and/or 2's. This problem has been thoroughly demolished.
2. Given a positive integer, R, find the smallest X such that 2^X contains R contiguous zeros. This problem has been solved, empirically, for values of R from 1 to 8, and it is known that for R = 9, $X \geq 60\ 000$. This problem remains baffling.

Back in issue number 9 (December 1973) we gave the results of some computer runs that indicate that for all $X > 169$, every value of F contains all the decimal digits, 0 through 9. This is certainly a plausible notion, and it has been verified for all values of X up to 20,000. At X = 20000, F contains 6021 digits, distributed as follows:

0	614
1	615
2	583
3	622
4	589
5	598
6	586
7	615
8	610
9	589

(Applying the frequency test of random digits to this data, we have a chi-squared value of 3.184, for a value of $p \cong .97$.) It has not been proved, however, that there could not be some high power of 2 that would lack one or more decimal digits.

Now along comes Richard Andree with a new set of research problems that includes this:

> Mathematicians have shown that, given a sequence of digits $S = d_1 d_2 d_3 \ldots d_n$, there exists an integer N such that 2^N begins with the sequence S.
>
> Produce a table giving S and N for all S from 1 to 100 such that N(S) is the <u>smallest</u> positive integer such that $2^{N(S)}$ begins with S. I think the first few values are:

S	Smallest N > 0
1	4
2	1
3	5
4	2
5	9
6	6 *
7	56
8	3
9	53
10	10
11	50

[It is not clear whether or not Professor Andree was being intentionally misleading, but the entry for 7 given in his list is incorrect.]

Results for S = 1 to 100 are given (for test purposes) in Table T.

The data of Table T (and extended for values of S up to 1000) can be obtained by generating powers of 2 and tabulating the 3 high-order digits. Flowchart F suggests one way of doing this, on a one-digit-per-word basis, suitable for the 8-bit personal machines.

Up to X = 1000, all values of S between 1 and 279 appear. For higher values of S, results get scarcer. Table G shows the distribution of the 2 high-order digits of 600 powers of 2 with X greater than 500. It can be seen from Table G that it will be difficult to extend Table T much beyond S = 400.

T

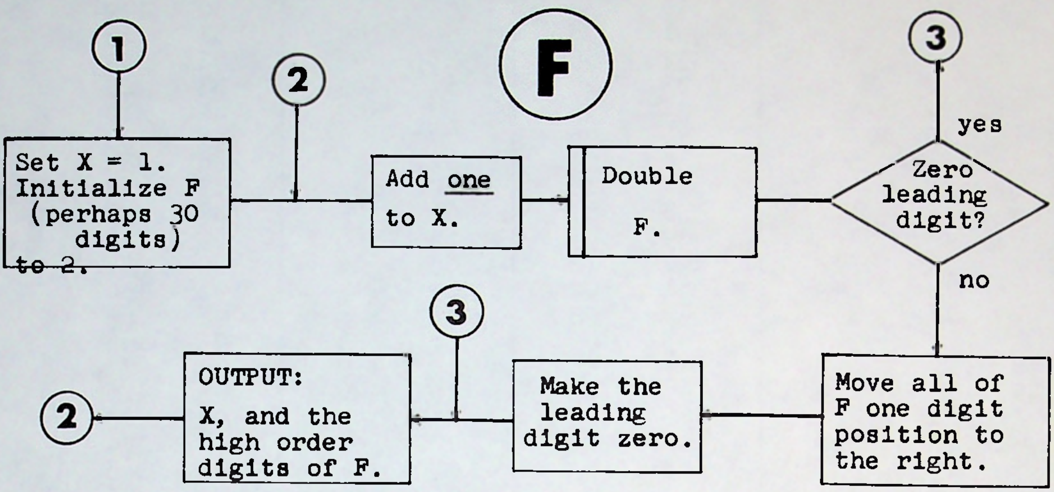
S	N	S	N	S	N	S	N
1	4	26	18	51	9	76	179
2	1	27	38	52	19	77	86
3	5	28	48	53	29	78	189
4	2	29	68	54	39	79	96
5	9	30	78	55	142	80	199
6	6*	31	98	56	49	81	13
7	46	32	5	57	59	82	209
8	3	33	25	58	162	83	23
9	53	34	35	59	69	84	219
10	10*	35	45	60	79	85	33
11	50	36	55	61	89	86	229
12	7	37	75	62	192	87	43
13	17	38	85	63	99	88	239
14	47	39	95	64	6	89	146
15	77	40	12	65	16	90	53
16	4	41	22	66	119	91	156
17	34	42	32	67	26	92	63
18	54	43	42	68	36	93	166
19	84	44	145	69	139	94	73
20	11	45	52	70	46	95	176
21	31	46	62	71	149	96	83
22	51	47	72	72	56	97	279
23	61	48	82	73	66	98	186
24	81	49	92	74	169	99	93
25	8	50	102	75	76	100	196

N is the smallest power of 2 having the digits of S as its high-order digits.

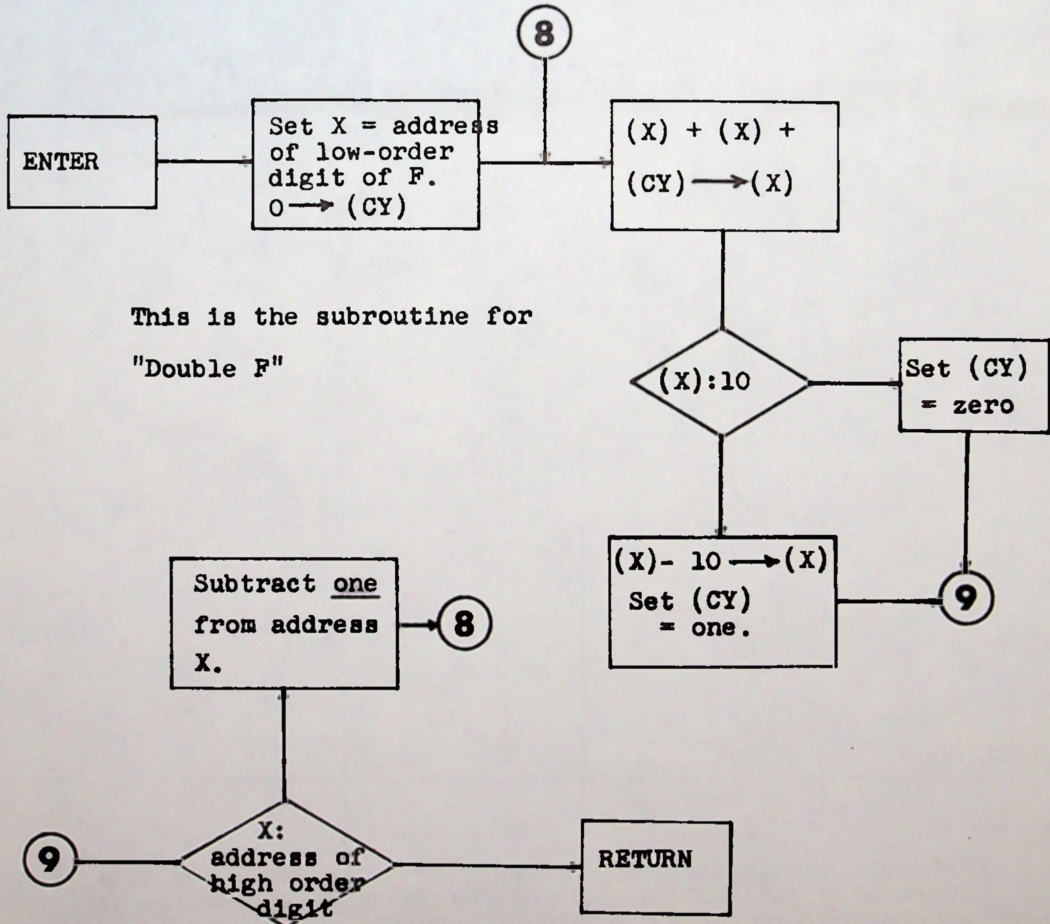
G

S	f	S	f	S	f	S	f
10	26	33	8	56	5	79	4
11	20	34	7	57	5	80	2
12	21	35	8	58	4	81	6
13	19	36	7	59	3	82	3
14	17	37	7	60	8	83	4
15	17	38	7	61	3	84	2
16	17	39	6	62	4	85	4
17	15	40	7	63	5	86	2
18	14	41	7	64	3	87	4
19	13	42	6	65	5	88	2
20	15	43	7	66	4	89	3
21	14	44	5	67	4	90	3
22	9	45	4	68	5	91	1
23	11	46	6	69	2	92	5
24	12	47	5	70	6	93	1
25	8	48	6	71	3	94	4
26	10	49	6	72	3	95	1
27	8	50	4	73	3	96	4
28	9	51	4	74	4	97	3
29	7	52	6	75	5	98	3
30	10	53	5	76	2	99	2
31	8	54	5	77	5		
32	9	55	3	78	2		

For each combination, S, of two high-order digits of some power of 2, column f shows the frequency of appearance in 600 powers, each above the 500th power.



This is the subroutine for "Double F"



But Professor Andree points out that the phenomenon described above for powers of 2 is also true for the powers of any base (other than base 10, or 100, and so on). Thus we have the start of more excellent computing problems. Tables S give the first 27 entries to these new tables, which should be extended.

The search for the high-order-digit sequences of Andree's problem suggests yet another powers-of-2 problem, and this one can be done in BASIC.

Since the sequences of digits in high powers of 2 are more or less random, then it follows that any sequence:

$$S = d_1 d_2 d_3 \ldots d_n$$

can be found somewhere <u>in</u> a power of 2, and a rather low power at that.

Casual inspection of a table of powers of 2 provides the beginning of a table:

S	N
1	4
2	1
3	5
4	2
5	8
6	4
7	15
8	3
9	12
10	10*

H

That is, the first appearance (that is, the one with the smallest exponent) of the sequence of digits S is in the power N. We wish to extend Table H indefinitely.

PROBLEM 268

One way to go about it is outlined in Flowchart J, and its subroutine, Flowchart K. The calculation is described in terms of values of S from 11 to 99, but the logic is readily altered to handle 3-digit values of S. (Note: The "Table T" referred to in Flowchart J is internal to that flowchart; it is not the same as the first table of results in this article. In Flowchart J it refers to a table in storage of 99 terms, with each term holding one digit of F. With this constraint, the value S = 200 will cause the program to reach Reference 5.)

3

1	9	13	38	25	26
2	3	14	15	26	47
3	1	15	13	27	3
4	14	16	34	28	24
5	10	17	11	29	45
6	8	18	32	30	66
7	6	19	9	31	22
8	4	20	30	32	43
9	2	21	7	33	108
10	21	22	28	34	20
11	19	23	49	35	85
12	17	24	5	36	41

5

1	3	10	63	19	9
2	2*	11	23	20	82
3	5	12	3	21	52
4	11	13	56	22	42
5	1	14	26	23	22
6	14	15	6	24	12
7	7*	16	69	25	2
8	50	17	49	26	75
9	10	18	29	27	55

7

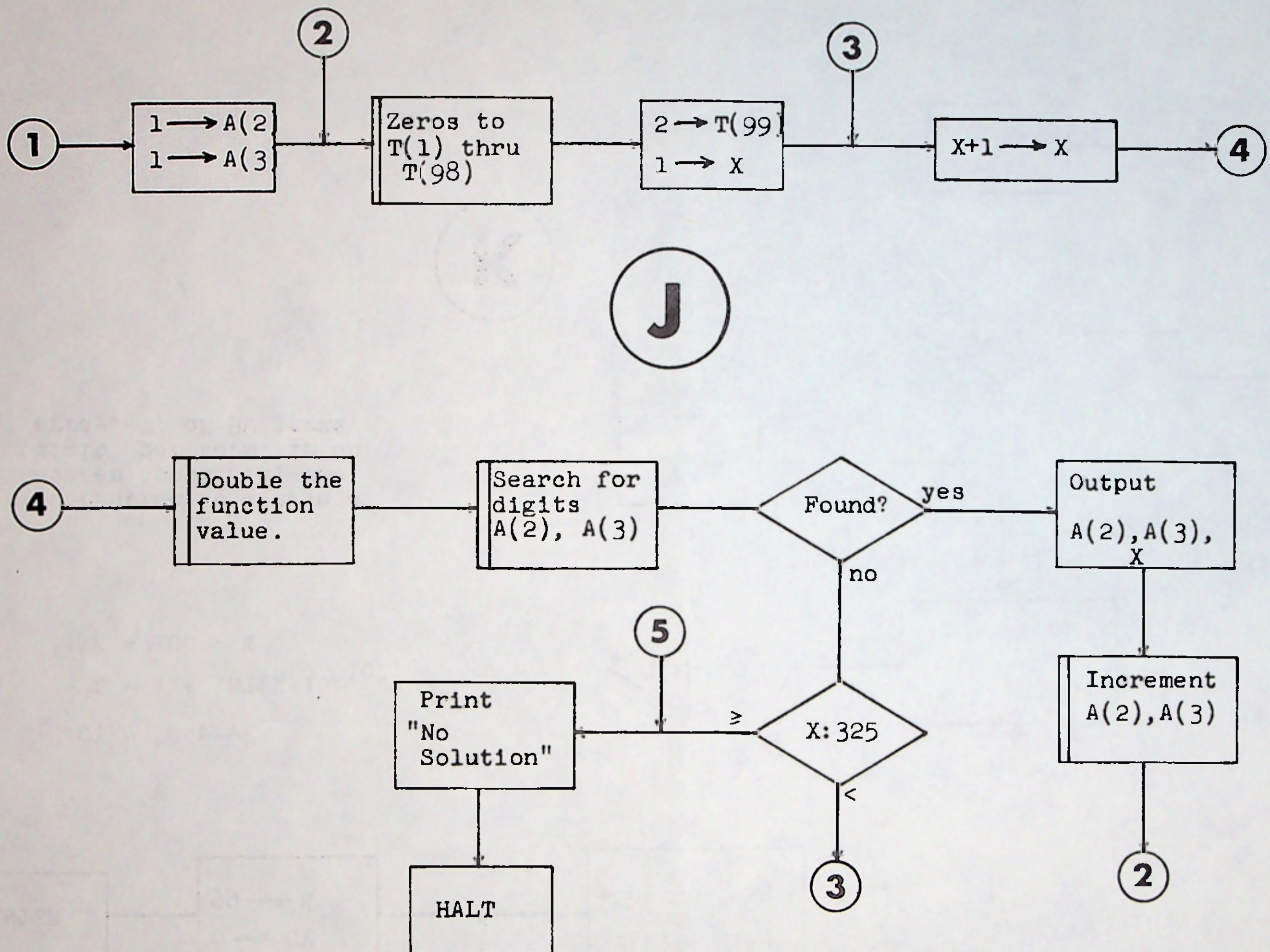
1	5	10	45	19	11
2	4	11	6	20	69
3	3*	12	38	21	43
4	9	13	12	22	30
5	8	14	57	23	17
6	14	15	31	24	4
7	1	16	5	25	49
8	7	17	50	26	36
9	13	18	37	27	23

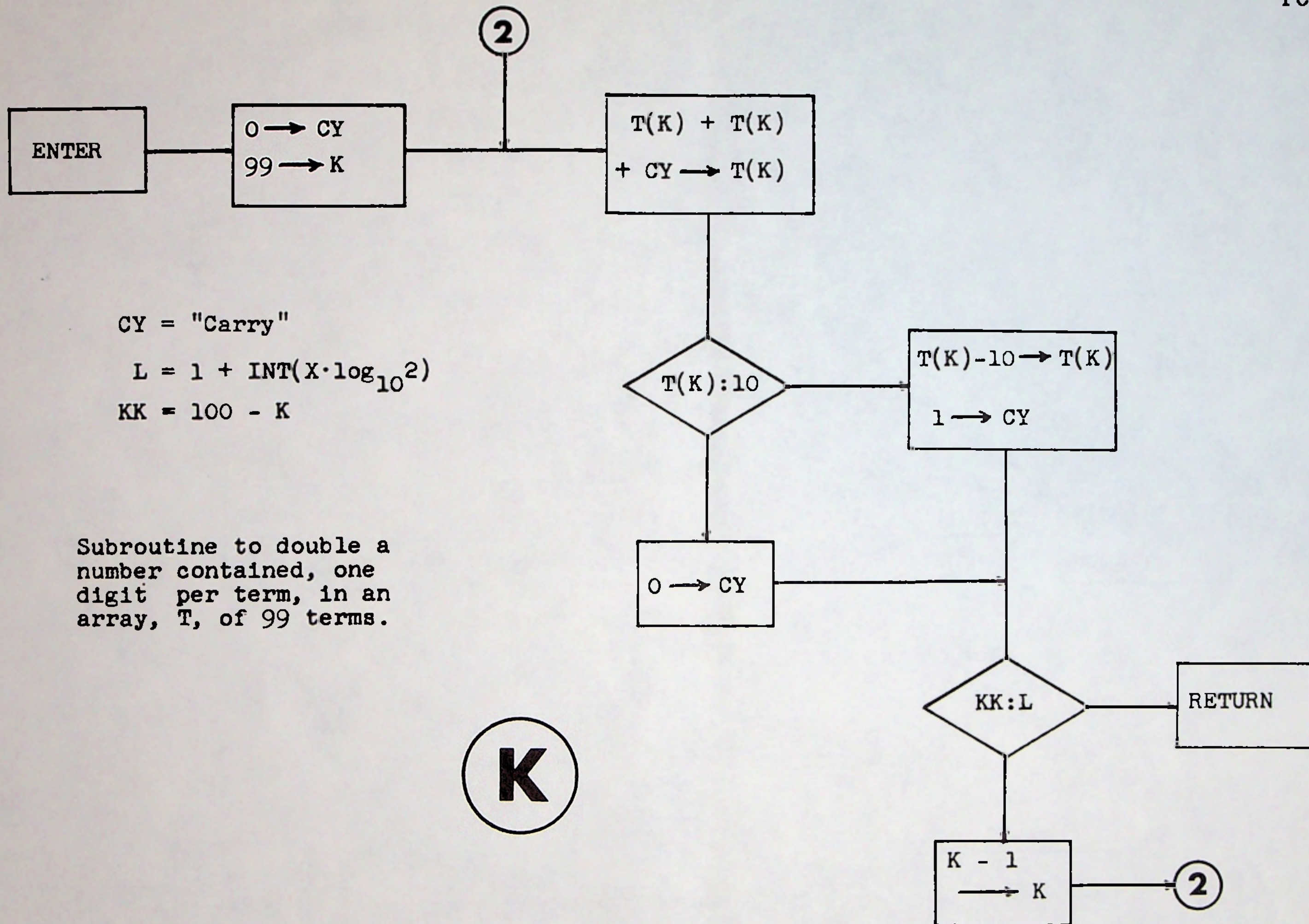
11

1	1*	10	25	19	7
2	8	11	1	20	56
3	12	12	2	21	8
4	15	13	3	22	57
5	17	14	4	23	9
6	19	15	29	24	82
7	21	16	5	25	10
8	22	17	6	26	118
9	24	18	55	27	59

S

For powers of 3, 5, 7, and 11 respectively, these tables show the appearance of the lowest power having S = 1, 2, 3,...,27 for their high-order digits.


1
1 → A(2
1 → A(3
2
Zeros to T(1) thru T(98)
2 → T(99)
1 → X
3
X+1 → X
4
J
4
Double the function value.
Search for digits A(2), A(3)
Found?
yes
no
Output A(2),A(3), X
Increment A(2),A(3)
2
X: 325
≥
<
3
5
Print "No Solution"
HALT



CY = "Carry"

L = 1 + INT$(X \cdot \log_{10} 2)$

KK = 100 - K

Subroutine to double a number contained, one digit per term, in an array, T, of 99 terms.

K

Going in another direction, and again treating the powers of 2 as a collection of more-or-less random digits, it must be possible to find sets of adjacent digits that sum to any value S, and Table M shows the start of a table of such powers. Thus, the 20th power of 2:

$$2^{20} = 1\ 0\ \underline{\underline{4\ 8\ 5\ 7\ 6}}$$

is the lowest power for which a set of adjacent digits sum to 30 (the first underscore) or to 31 (the longer underscore). The entries in Table M are somewhat predictable. For example, for S = 100, the power of 2 must be high enough to provide sufficient digits. With random non-zero digits averaging 5.0, it would take around 20 digits to sum to 100, and the smallest power of 2 that has 20 digits in it is the 64th. As it turns out, the lowest power containing a set of adjacent digits that sum to 100 is the 67th.

And, of course, the same reasoning leads to Table N, for products of any size formed from adjacent groups of digits in powers of 2.

1	4	13	8	25	16
2	1	14	11	26	15
3	5	15	15 *	27	23
4	2	16	16 *	28	24
5	5 *	17	14	29	19
6	4	18	13	30	20
7	4	19	12	31	20
8	3	20	13	32	28
9	12	21	14	33	23
10	6	22	14	34	26
11	7	23	15	35	27
12	11	24	19	36	24

M

The smallest power of 2 for which sums of adjacent digits form each value of S.

1	40	14	15	27	43
2	7	15	16	28	18
3	17	16	7	29	unkn
4	18	17	34	30	8
5	9	18	13	31	17
6	4	19	unkn	32	11
7	21	20	30	33	unkn
8	10	21	30	34	27
9	13	22	50	35	20
10	8	23	unkn	36	35
11	42	24	6	37	37 *
12	18	25	16	38	13
13	17	26	49	39	unkn

N

The smallest power of 2 for which the product of groups of adjacent digits form each value of S.

Curious: In every table that we have constructed, there is at least one entry for which S = N; we have marked these entries with (*). (For the table of the leading digits of powers of 3 there is no marked entry, inasmuch as the first instance in which S = N occurs at 185.)

Could it be that any such table will have an entry for S = N?

Multiple Choice Tests

A multiple choice test is given in which multiple correct answers are possible for any question.

The accompanying diagram shows the situation for a 10-question test; the cross-hatched squares are the correct responses. For such a test, there are four possibilities for each square of the pattern:

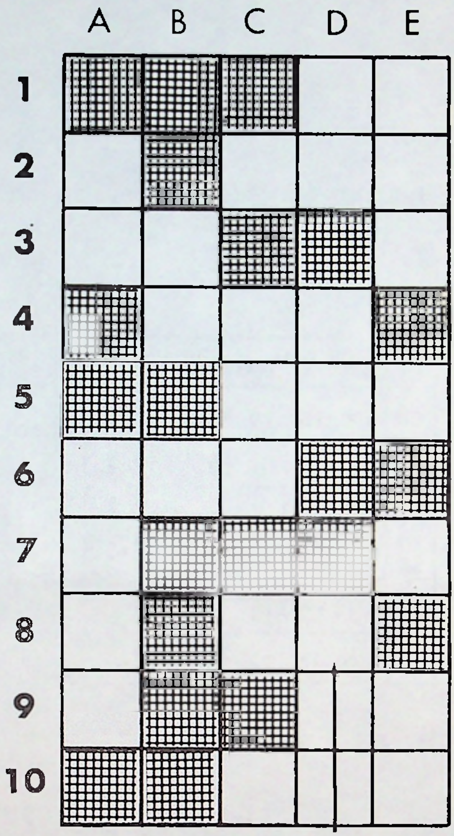
Correct response	X	X		
Student response	X		X	
Score	+2	-1	-2	0

If the student reponses are marked on a printed grid of squares, then grading can be facilitated by cutting a cardboard template. The squares that are cross-hatched would be cut out, to reveal the places where X's should be made.

But now there is a problem. Suppose that response 8D is also correct? There will be an unwelcome hole at 8C in the template. In this case, interchanging questions 9 and 10 would solve the problem. In general, a series of questions might have to be carefully rearranged in order to permit the cutting of a proper template. Specifically, what is the logic of a program that will accept input like the following and output an ordering of the question numbers such that a template could be cut?

1. ACE	7. BDE	13. CDE	19. CDE	25. BDE
2. BDE	8. ACDE	14. CE	20. E	26. CDE
3. AB	9. BC	15. AE	21. AB	27. BD
4. CDE	10. BCDE	16. ADE	22. AC	28. ACE
5. BC	11. BE	17. BCE	23. AE	29. ACD
6. BE	12. BDE	18. BCD	24. ABC	30. BDE

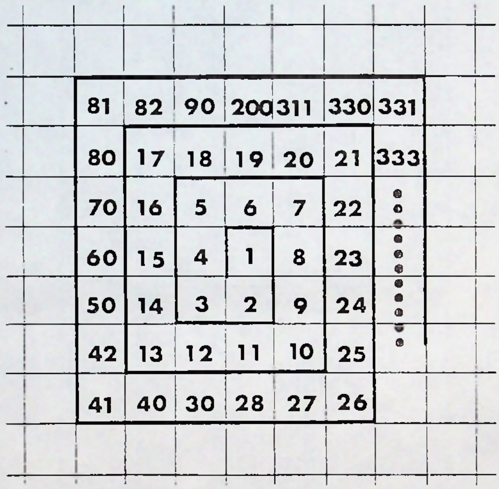
PC83-13



The plan for a portion of a template to use in grading a test. Each question of the test has multiple correct answers. The test answer sheet is made to conform to the template, so that the template reveals correctly marked squares as well as squares that have been marked in error. The arrow points to a problem area in constructing such a template.

Problem Solution

In issue number 66 there appeared this problem (Number 240):



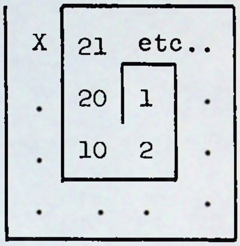
The Chicago Loop Trip

The consecutive integers are inserted into the spiral of squares, subject to just one constraint; namely, that a number will not be used if it shares a digit in common with a previous number in the square orthogonal to it. Thus, the numbers proceed normally until 29, which is skipped because of the common 2 it has with 12.

The Figure shows the first 44 numbers, and Problem 240 called for extending the calculation. No one has yet done this.

Meanwhile, however, there is a communication from Harry L. Nelson (Executive Editor of the Journal of Recreational Mathematics and co-discoverer of the largest known prime, a Mersenne prime with the exponent 44497):

"...I thought it might terminate. That is, it might reach a point where no digits are left to be used for the next entry. This <u>can</u> happen in base 3:



...it is obvious that when we get to X, no value can go there. I think I can prove that this will never happen in any base larger than three, but not by computer."

While he was at it, Mr. Nelson also did some work on Problem 242, also in issue 66:

$$a_0 = 0$$

$$a_1 = 1$$

$$a_{n+1} = a_n - \left[(1/2)(a_n + 1)\right]$$

unless that number has occurred earlier, in which case the sign changes from - to +. The square brackets denote "greatest integer in."

The problem comes from Dave Silverman. Mr. Nelson calculated the first 10,000 terms of the sequence; he has proved that the sequence is non-terminating, but cannot yet prove that it will eventually include every integer. In the first 10,000 terms, the largest number that appears is 531,440 and the smallest number that does not appear is 2408.

Problem Solution

Fred Way, Case Western Reserve University, offers a solution to Problem 262 ("Currency Exchange") which appeared on the cover of issue number 80.

The problem postulated a stable relation between the leading currencies of the world. The stable condition is suddenly perturbed by a demand for Pesos in terms of Dollars, such that their price goes up from \$.0438 to \$.0472 with no other change in currency exchange rates. How could one capitalize on that situation?

Mr. Way says:

"There will be a change of -12.27% in the value of the French Franc relative to the Peso. Borrow 1000 Francs and buy 5312.8 Pesos. Wait for the news to get out (that is, for the change in values to be reflected across the complete table of exchange rates). Then use 4661 Pesos to buy 1000 Francs, pay off the loan, and have 651.8 Pesos profit."

Mr. Way performed all his calculations in APL, in which he can generate complete new tables like that on the cover of issue 80 with one command:

```
⍉ V ∘ . ÷ V
```

Old Timer's Quiz

1. What do the letters KSNJFL stand for?
2. Define "echo check."
3. What is progressive digiting?
4. What is block sorting?
5. What was the last commercial computer that had two instructions per word?
6. Who invented index registers?
7. Who invented closed (i.e., linked) subroutines?
8. What was the word size of the following machines: (a) IBM 650; (b) LGP-30; (c) IBM 1620; (d) IBM 701?
9. What happened at Louisville?
10. What are short and long hammerlocks?
11. What is offset master gang punching?
12. What is "Plug to C"?
13. What is a skip bar?
14. What is a hopper stop switch?
15. How many different computers had variable word lengths?
16. How did a Table Lookup op-code function?
17. What is an emitter?
18. What is a collating device for a sorter?
19. Distinguish between pilot selectors, latch selectors, digit selectors, and co-selectors.
20. What is a back circuit?

PROBLEM 271

<u>SRA Data Processing Glossary</u>

(with concepts written by Robert C. Malstrom)
Science Research Associates, 1979, 8 1/2 x 11,
soft cover, 281 pages.

The last glossary that bore the imprimatur of a professional group was the 1954 glossary produced by ACM, with a committee headed by Grace Hopper. It was a thin little thing--how much was there to define, in the year that the first computer was to be used on a business application?

Since then there have been dozens of glossaries produced (by governmental agencies, by manufacturers, and by private individuals), but our industry has no official (or even quasi-official) glossary. It is difficult to write a contract that will hold up in court without some wasy of falling back on official definitions of complex technical terms. Consider, for example, a contract with a software house to produce a compiler. The buyer doubts that what is delivered is really a compiler. How could this ever be settled in court if one had to rely on the definition of <u>compiler</u> that is given below?

Still, there is plenty of room for any new glossary, such as this one. The entries are largely taken from other sources (e.g., the <u>American National Dictionary for Information Processing</u>, copyright 1977 by CBEMA).

Malstrom prefaces the book with 22 pages of discussion of computing and data processing, so that the terms in the glossary do not appear from nowhere, but are put somewhat into context--a splendid idea.

The book is nicely produced, and is typeset. It contains some 5000 entries.

There are only two tests of a glossary; namely, does it have the words I want defined or explained? and are the definitions clear and acceptable? The first part seems to have been satisfied by the sheer bulk of the book. The second part can be judged by sampling a few entries. Comments by the reviewer are in parentheses.

Book Review ...

Compiler. A compiler program used to compile.
Synonymous with compiling program.

Compiling program. Synonym for compiler.
(The book has much of this circularity.)

Compile. To translate a computer program expressed in a problem-oriented language into a computer-oriented language.

Interpreter. A computer program used to interpret.
Synonymous with interpretive program.
(Care to guess what you will find under "Interpretive program"?)

Random access storage. Deprecated term for direct access storage.
(Deprecate means "to express disapproval of.")

Direct access storage. A storage device that provides direct access to data. See also immediate access storage.

Immediate access storage. A storage device whose access time is negligible in comparison with other operating times.
(With the above definitions, I do not see how one could discriminate between cores, tapes, drums, and disks, insofar as their access times were concerned.)

System parameter record (SPR). A user-built record in the 3660 Supermarket Store System that selects the operation sequence of a store controller and terminals.
(The book contains many such parochial terms.)

Cold start. In VM/370, a system restart...
(The term "cold start" was in general use two decades before the IBM 370.)

Some quotes from the SRA glossary, and comments by the reviewer.

ANNUAL INDEX

Volume 7 1979
Issues 70 through 81
240 pages